Current Status of Utah's C and Runtime System for the DECSystem 20

Last revised 4/25/83

CAVEAT:
THIS IS NOT A DISTRIBUTION. This is only a "version." It is essentially a "day-in-the-life-of."

DOCUMENTATION:
The compiler description provided here is, as far as we know, complete and accurate. On the other hand, documentation of the library routines is minimal; possession of the Unix Programmer's Manual is assumed. In general, limitations of the library routines are noted only in this document or the source and are not always complete.

COMPILER:
The compiler is based on the Portable C Compiler (PCC) from V7/32V UNIX*. It accepts the complete language as defined in "The C Programming Language" by Kernighan and Ritchie (Prentice-Hall, 1978), with the additional features of the V7 compiler (enumerated types and structures as function arguments and values.) The generated code is not wonderful but not bad either. There are no known instances of incorrect code generation and the compiler has been stable for a considerable length of time. The current semantic limitations are:

PREPROCESSOR (cpp) and PASS1 (c1)
1. #include file specifications must be in unix format. (All file specs referenced by the runtime library can be either unix or tops20 format.) See <pcc.doc>misc.doc for filename format details.

2. A bracketed #include filename may not contain a digit.

PASS2 (c2)
1. Currently, chars == shorts == ints == longs == unsigneds == 1 word. The next version of the compiler will provide packed characters.

2. Doubles == floats == 1 word. All floating point arithmetic is carried out in single precision.

3. Unsigneds are treated as ints. In general, it doesn't matter for porting code from 11s or Vaxen, as 32 << 36. A future version will do unsigned arithmetic properly.

4. Fancy structure operations have not been fully tested but are not known to fail.
   a. Simple structure assignments, "s1 = s2;" definitely work fine. More complex exprs e.g., "if (s1 = s2) ..." might bomb.
   b. Structures as args to functions and as values returned have not been fully tested.

5. As with all pcc implementations, some syntax errors generate "compiler error" error messages. Occasionally a not so complex expression, usually involving a function call, proves too complicated for the compiler and must

be simplified. In practise this is an insignificant problem.

ASSEMBLER/LINKER
The standard DEC assembler and linker are used. A PCL procedure avoids the filename restrictions of LINK's command lanmguage.

1. External identifiers are unique only to 6 chars, case-folded.
2. Multiple *definitions* of an identifier in different modules don't work. All but one must be declared "extern."
3. Certain identifiers, if used in a particular way (usually as an address constant), may not work due to collision with an assembler pseudo-op. This is very rare.

TOP-LEVEL DRIVING PROCEDURE
As distributed, a complex and wonderful PCL procedure runs the show, but many versions of PCL are quite buggy and this might be a problem at some sites. Some bug fixes to an older version PCL are included on the tape If you do not have PCL a simple MACRO driving program would be easy to write, and essential to achieving reasonable convenience.

RUN TIME SYSTEM:
Command line input/output redirection and filename expansion are supported. The following system and library calls are implemented in the current version of the library; the future "release" will implement nearly all section 2 system calls and section 3 library routines.

*** SECTION 2 SYSTEM CALLS ***

open, close	additional modes supported: append, exclusive open, arbitrary byte size, default (existing) byte size. See <file.h> for optional 3rd param. Opens currently default to 7 bit.
read, write	On 7-bit i/o CRs are stripped/added invisibly to user pgm.
execl, execv	
fork, wait	fork/shared fd semantics not exact; also fork is slow now...
exit, _exit	these are the same. Exit status goes to reg 0 of parent process
signal, kill, alarm	Crude, preliminary implementation of signal. Has known bug when input ^C\r in CBREAK mode. sigquit maps to ^B, not ^\. These are supported to varying degrees: (hup),int,quit,ill,(kill),fpe,(bus,segv),alrm,term
brk, sbrk	
stat, fstat	all fields mapped except nlink, gid, rdev.
ioctl	SETP, SETN, GETP, with control of ECHO,CBREAK,RAW,CRMOD,speeds
time	true Unix internal format
access	only mode 0 currently implemented
chdir	optional password arg
creat	currently ignores mode
unlink	optional additional arg allows options, e.g. expunge.
getuid	
utime	
abort	
errno, _errno	errno is set to last Tops-20 error code w/o the 0600000 bits. _errno retains the 0600000.

```
*** USE WITH CAUTION ON NON-BINARY FILES DUE TO CR/LF PROBLEM ***
lseek, tell             i.e, beware of relative seeks on text files
fseek, ftell, rewind

*** SECTION 3 I/O ***
fopen, freopen, fdopen   read/write, append, "binary" modes also supported
        via Berkeley + convention, 'a' == append, 'b' = binary, e.g.
        fopen(foo, "r+"); open(foo, "wb");  fopen(bar, "a");
fclose, fflush
fread, fwrite
printf, fprintf, sprintf, _doprnt       user-extensible format types
scanf, fscanf, sscanf                   sscanf not well tested, may be buggy
gets, fgets
puts, fputs
setbuf
isatty
ungetc
getc, fgetc, getchar, getw
putc, fputc, putchar, putw
clearerr, feof, ferror, fileno

*** SECTION 3 STRING ROUTINES ***
index, rindex, strcat, strcmp, strcpy, strlen, strncat, strncmp, strncpy

*** SECTION 3 MISC ***
malloc, calloc, realloc, valloc, free
all of Termcap and Curses libraries - Curses lib is separate in pcc:curses.rel
ctype macros
atoi, atol, atof
setjmp, longjmp
ctime, asctime, gmtime, localtime, timezone
perror
qsort
mktemp                  Uses the "high-precision clock" for its random number.
getenv
getlogin
getpwuid
re_comp, re_exec
crypt, encrypt, setkey
nargs

*** SECTION 3 MATH ***
sin, cos
modf, frexp
ecvt, fcvt
floor, ceil

*** LOCAL/CMU STRING(3) ROUTINES ***
sindex, rsindex
skipto, skipover
fold, foldup, folddown
stlmatch, stablk, prstab
nxtarg
getbool
```

*** TOPS-20 ROUTINES ***
The following routines do not have Unix analogues, but instead provide convenient and direct access to TOPS-20 monitor facilities. The inclusion of the JSYS function allows programmers to write "native mode" C code which directly accesses TOPS-20 functions instead of using the Unix emulation facilities of the rest of the library.

```
jsys(JSnum, acs)                Grovel in the opsys!
jsys.h, monsym.h, monerr.h      MONSYM translated to C include format
fnstd, fnunix           Convert filenames between Unix and Tops-20 formats
_glob                   Expand filenames
_gtjfn, _rljfn, _gnjfn
_gtfsz
_jfns
_seterr()
_chfdb
```

UNIX PROGRAMS CURRENTLY PORTED
Many more could be easily or trivially ported (just compile them); these are the ones we've so far had need for here that have been easy. (Non-source tapes include only those programs marked with an asterisk.) In general, source for only the non-standard programs is on the tape, even on "source" tapes, as the code is pretty much identical to that of v7 or 4.1bsd.

ps:<unix>*.exe, 41 files.

ar	comm	diff	hangman	more*	printenv*	sleep	uniq
asmlist*	cr_ar	echo	kc*	num	psplit*	touch	uudecode*
cal	date	egrep	lex	od	rogue*	treset*	uuencode*
cb	dc	expand	life*	pr	sed	ul	wc
cdecl*	ded*	grep	mille*	prfmt*	setdate*	unexpand	yacc
cmp							

ar - not all options work
asmlist - Merges assembler output from compiler with C source
cdecl - Useful C utility from Usenet; man page on <unix.man>
ded - Directory editor, same as "dired" posted on Usenet
diff - is the restricted version of diff normally invoked by "diff -h"
kc - kill (strip) carriage returns from a file
od - doesn't necessarily work right for many arg types
rogue - Version 3.6

*UNIX is a trademark of Bell Laboratories